Recognizability and Model-Checking from Automata to Lambda-Calculus

Sylvain Salvati
sylvain.salvati@labri.fr, INRIA
Schematology

Programs

Verification
Typing/Logic/... Semantics

Execution
Schematology

Programs

- Verification
- Execution

Typing/Logic/... Semantics

Adequacy
Schematology

Programs

Schemes

Evaluation trees

Typing/Logic/…

Semantics

Verification

Execution

Syn. abst.

Syn. exec.

Adequacy
Schematology

Programs

Schemes

Evaluation trees

Typing/Logic/…

Verification

Syn. abst.

Execution

Syn. exec.

Evaluation

Semantics

Syn. prop.

Evaluation

Adequacy
Schemes and MSOL

- MSOL allows one to specify how traces of programs should look like.
Schemes and MSOL

- MSOL allows one to specify how traces of programs should look like.
- We shall see in this course that the MSOL theory of a wide class of program schemes is decidable.
Schemes and MSOL

- MSOL allows one to specify how traces of programs should look like.
- We shall see in this course that the MSOL theory of a wide class of program schemes is decidable.
- This wide class of schemes are simply typed $\lambda Y$-terms built on a tree signature:
Schemes and MSOL

- MSOL allows one to specify how traces of programs should look like.
- We shall see in this course that the MSOL theory of a wide class of program schemes is decidable.
- This wide class of schemes are simply typed $\lambda Y$-terms built on a tree signature:
  - They define a rich control flow.
Schemes and MSOL

- MSOL allows one to specify how traces of programs should look like.
- We shall see in this course that the MSOL theory of a wide class of program schemes is decidable.
- This wide class of schemes are simply typed $\lambda Y$-terms built on a tree signature:
  - They define a rich control flow.
  - They model higher-order programs: here MSOL can model safety properties and resource usage.
Schemes and MSOL

- MSOL allows one to specify how traces of programs should look like.
- We shall see in this course that the MSOL theory of a wide class of program schemes is decidable.
- This wide class of schemes are simply typed $\lambda Y$-terms built on a tree signature:
  - They define a rich control flow.
  - They model higher-order programs: here MSOL can model safety properties and resource usage.
  - They can model reactive systems: here MSOL can model communication pattern, fairness and liveness properties.
**Types** 0 is a type and \((A \rightarrow B)\) is a type if \(A\) and \(B\) are types. The set of types is noted \text{types}.

**Tree signature** \(\Sigma = \{a, b, \ldots\}\) all constants of type \(0 \rightarrow 0 \rightarrow 0\) or of type 0.

**\(\lambda Y\)-calculus**

The indexed family of typed terms \(\left(\Lambda_A\right)_{A \in \text{types}}\) is defined inductively as:

- the typed variable \(x^A\) is in \(\Lambda_A\),
- if \(a\) is a constant, then its type is \(0 \rightarrow 0 \rightarrow 0\),
- if \(M\) is in \(\Lambda_{B \rightarrow A}\) and \(N\) in \(\Lambda_B\), then \((MN)\) is in \(\Lambda_A\),
- if \(M\) is in \(\Lambda_A\) then \((\lambda x^B.M)\) is in \(\Lambda_{B \rightarrow A}\),
- for every \(A\), \(Y^A\) is in \(\Lambda_{(A \rightarrow A) \rightarrow A}\),
- for every \(A\), \(\omega^A\) is in \(\Lambda_A\).

**Typing à la Church**: every term has a unique type
Notational conventions

- $M_0 M_1 \ldots M_n$ denotes $\ldots (M_0 M_1) \ldots M_n$
- $\lambda x_1 \ldots x_n.M$ denotes $\lambda x_1 \ldots ((\lambda x_n.M) \ldots)$
- $\lambda x_1 \ldots x_n.M_0 M_1 \ldots M_n$ denotes $\lambda x_1 \ldots ((\lambda x_n.(\ldots (M_0 M_1) \ldots M_n) \ldots) \ldots)$
Capture avoiding substitution $M[N/x]$: 

- $y[N/x] = x$ when $y \neq x$ and $x[N/x] = N$,
Capture avoiding substitution

Capture avoiding substitution $M[N/x]$:

- $y[N/x] = x$ when $y \neq x$ and $x[N/x] = N$,
- $(\lambda y. M)[N/x] = (\lambda y. M[N/x])$ when $y \not\in \text{FV}(N)$; in case $y \in \text{FV}(N)$, we need to change the rename the variable $y$ (and its free occurrences in $M$) into a fresh variable,
Capture avoiding substitution $M[N/x]$:

- $y[N/x] = x$ when $y \neq x$ and $x[N/x] = N$,
- $(\lambda y. M)[N/x] = (\lambda y. M[N/x])$ when $y \notin FV(N)$; in case $y \in FV(N)$, we need to change the rename the variable $y$ (and its free occurrences in $M$) into a fresh variable,
- $(M_1 M_2)[N/x] = (M_1[N/x] M_2[N/x])$, 

- $y[N/x] = x$ when $y \neq x$ and $x[N/x] = N$, 
- $(\lambda y. M)[N/x] = (\lambda y. M[N/x])$ when $y \notin FV(N)$; in case $y \in FV(N)$, we need to change the rename the variable $y$ (and its free occurrences in $M$) into a fresh variable,
- $(M_1 M_2)[N/x] = (M_1[N/x] M_2[N/x])$, 

- $y[N/x] = x$ when $y \neq x$ and $x[N/x] = N$, 
- $(\lambda y. M)[N/x] = (\lambda y. M[N/x])$ when $y \notin FV(N)$; in case $y \in FV(N)$, we need to change the rename the variable $y$ (and its free occurrences in $M$) into a fresh variable,
- $(M_1 M_2)[N/x] = (M_1[N/x] M_2[N/x])$, 

- $y[N/x] = x$ when $y \neq x$ and $x[N/x] = N$, 
- $(\lambda y. M)[N/x] = (\lambda y. M[N/x])$ when $y \notin FV(N)$; in case $y \in FV(N)$, we need to change the rename the variable $y$ (and its free occurrences in $M$) into a fresh variable,
- $(M_1 M_2)[N/x] = (M_1[N/x] M_2[N/x])$, 

- $y[N/x] = x$ when $y \neq x$ and $x[N/x] = N$, 
- $(\lambda y. M)[N/x] = (\lambda y. M[N/x])$ when $y \notin FV(N)$; in case $y \in FV(N)$, we need to change the rename the variable $y$ (and its free occurrences in $M$) into a fresh variable,
- $(M_1 M_2)[N/x] = (M_1[N/x] M_2[N/x])$, 

- $y[N/x] = x$ when $y \neq x$ and $x[N/x] = N$, 
- $(\lambda y. M)[N/x] = (\lambda y. M[N/x])$ when $y \notin FV(N)$; in case $y \in FV(N)$, we need to change the rename the variable $y$ (and its free occurrences in $M$) into a fresh variable,
- $(M_1 M_2)[N/x] = (M_1[N/x] M_2[N/x])$, 

- $y[N/x] = x$ when $y \neq x$ and $x[N/x] = N$, 
- $(\lambda y. M)[N/x] = (\lambda y. M[N/x])$ when $y \notin FV(N)$; in case $y \in FV(N)$, we need to change the rename the variable $y$ (and its free occurrences in $M$) into a fresh variable,
Capture avoiding substitution $M[N/x]$:

- $y[N/x] = x$ when $y \neq x$ and $x[N/x] = N$,
- $(\lambda y. M)[N/x] = (\lambda y. M[N/x])$ when $y \not\in FV(N)$; in case $y \in FV(N)$, we need to change the rename the variable $y$ (and its free occurrences in $M$) into a fresh variable,
- $(M_1 M_2)[N/x] = (M_1[N/x] M_2[N/x])$,
### Operational semantics

A relation $R$ is a congruential relation over $\lambda Y$-terms if:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(M, N) \in R$</td>
<td>$M , R , N$</td>
</tr>
<tr>
<td>$M_1 , R , M_2$</td>
<td>$M_1 , N , R , M_2 , N$</td>
</tr>
<tr>
<td>$M_1 , R , M_2$</td>
<td>$N , M_1 , R , N , M_2$</td>
</tr>
</tbody>
</table>
| $\lambda x. M_1 \, R \, \lambda x. M_2$ | $\beta\text{-contraction}$
| $\eta\text{-contraction}$
| $\delta\text{-contraction}$
| $\beta\text{-reduction}$
| $\beta\text{-conversion}$
| $\beta\delta\text{-contraction}$
| $\beta\delta\text{-reduction}$
| $\beta\delta\text{-conversion}$ |
Operational semantics

A relation $\mathcal{R}$ is a congruential relation over $\lambda Y$-terms if:

$$(M, N) \in \mathcal{R} \quad \frac{M_1 \mathcal{R} M_2}{M \mathcal{R} N} \quad \frac{M_1 \mathcal{R} M_2}{M_1 N \mathcal{R} M_2 N} \quad \frac{M_1 \mathcal{R} M_2}{N M_1 \mathcal{R} N M_2} \quad \frac{M_1 \mathcal{R} M_2}{\lambda x. M_1 \mathcal{R} \lambda x. M_2}$$

- $\beta$-contraction $\rightarrow_\beta$ is the least congruential relation s.t.:

$$(\lambda x. M) N \rightarrow_\beta M[N/x]$$
Operational semantics

A relation $R$ is a congruential relation over $\lambda Y$-terms if:

\[
\begin{array}{cccc}
(M, N) \in R & M_1 R M_2 & M_1 R M_2 & M_1 R M_2 \\
M R N & M_1 N R M_2 N & NM_1 R NM_2 & \lambda x. M_1 R \lambda x. M_2 \\
\end{array}
\]

- $\mathbf{\beta}$-contraction $\rightarrow_{\beta}$ is the least congruential relation s.t.:
  \((\lambda x. M) N \rightarrow_{\beta} M[N/x]\)

- $\mathbf{\eta}$-contraction $\rightarrow_{\eta}$ is the least congruential relation s.t.:
  \(\lambda x. Mx \rightarrow_{\eta} M \text{ when } x \notin FV(M),\)
Operational semantics

A relation $\mathcal{R}$ is a congruential relation over $\lambda Y$-terms if:

$$(M, N) \in \mathcal{R} \quad M_1 \mathcal{R} M_2 \quad M_1 \mathcal{R} M_2 \quad \lambda x. M_1 \mathcal{R} \lambda x. M_2$$

- $\beta$-contraction $\rightarrow_\beta$ is the least congruential relation s.t.:
  $$(\lambda x. M) N \rightarrow_\beta M[N/x]$$

- $\eta$-contraction $\rightarrow_\eta$ is the least congruential relation s.t.:
  $$\lambda x. Mx \rightarrow_\eta M \text{ when } x \notin FV(M),$$

- $\delta$-contraction $\rightarrow_\delta$ is the least congruential relation s.t.:
  $$YM \rightarrow_\delta M(YM).$$
**Operational semantics**

A relation $R$ is a congruential relation over $\lambda Y$-terms if:

- $(M, N) \in R \Rightarrow M \mathrel{R} N$
- $M_1 \mathrel{R} M_2 \Rightarrow M_1 N \mathrel{R} M_2 N$
- $M_1 \mathrel{R} M_2 \Rightarrow NM_1 \mathrel{R} NM_2$
- $\lambda x. M_1 \mathrel{R} \lambda x. M_2$

- $\beta$-contraction $\rightarrow_\beta$ is the least congruential relation s.t.:
  - $(\lambda x. M) N \rightarrow_\beta M[N/x]$

- $\eta$-contraction $\rightarrow_\eta$ is the least congruential relation s.t.:
  - $\lambda x. M x \rightarrow_\eta M$ when $x \notin \text{FV}(M)$,

- $\delta$-contraction $\rightarrow_\delta$ is the least congruential relation s.t.:
  - $YM \rightarrow_\delta M(YM)$,

- $\beta$-reduction $\rightarrow^*_\beta$: reflexive and transitive closure of $\rightarrow_\beta$, 

$\beta\delta$-contraction $\rightarrow_{\beta\delta}$: union of $\rightarrow_\beta$ and $\rightarrow_\delta$, 

$\beta\delta$-reduction $\rightarrow^{*\ast}_{\beta\delta}$: reflexive and transitive closure of $\rightarrow_{\beta\delta}$, 

$\beta\delta$-conversion $=\rightarrow_{\beta\delta}$: reflexive, transitive and symmetric closure of $\rightarrow_{\beta\delta}$.
Operational semantics

A relation \( R \) is a congruential relation over \( \lambda Y \)-terms if:

\[
(M, N) \in R \quad M_1 R M_2 \quad M_1 R M_2
\]

- **\( \beta \)-contraction \( \rightarrow_\beta \)** is the least congruential relation s.t.:
  \[
  (\lambda x. M) N \rightarrow_\beta M[N/x]
  \]
- **\( \eta \)-contraction \( \rightarrow_\eta \)** is the least congruential relation s.t.:
  \[
  \lambda x. M x \rightarrow_\eta M \text{ when } x \notin \text{FV}(M),
  \]
- **\( \delta \)-contraction \( \rightarrow_\delta \)** is the least congruential relation s.t.:
  \[
  YM \rightarrow_\delta M(YM).
  \]
- **\( \beta \)-reduction \( \rightarrow^*_\beta \)**: reflexive and transitive closure of \( \rightarrow_\beta \),
- **\( \beta \)-conversion \( =_\beta \)**: reflexivity, transitive and symmetric closure of \( \rightarrow_\beta \),
Operational semantics

A relation $\mathcal{R}$ is a congruential relation over $\lambda Y$-terms if:

$$(M, N) \in \mathcal{R} \quad \frac{M_1 \mathcal{R} M_2}{M_1 M \mathcal{R} M_2} \quad \frac{M_1 \mathcal{R} M_2}{NM_1 \mathcal{R} NM_2} \quad \frac{M_1 \mathcal{R} M_2}{\lambda x. M_1 \mathcal{R} \lambda x. M_2}$$

- $\beta$-contraction $\rightarrow_\beta$ is the least congruential relation s.t.:
  $$(\lambda x. M)N \rightarrow_\beta M[N/x]$$
- $\eta$-contraction $\rightarrow_\eta$ is the least congruential relation s.t.:
  $\lambda x. Mx \rightarrow_\eta M$ when $x \notin FV(M)$,
- $\delta$-contraction $\rightarrow_\delta$ is the least congruential relation s.t.:
  $YM \rightarrow_\delta M(YM)$.

- $\beta$-reduction $\rightarrow^*_\beta$: reflexive and transitive closure of $\rightarrow_\beta$,
- $\beta$-conversion $\equiv_\beta$: reflexivity, transitive and symmetric closure of $\rightarrow_\beta$,
- $\beta\delta$-contraction $\rightarrow_{\beta\delta}$: union of $\rightarrow_\beta$ and $\rightarrow_\delta$. 
Operational semantics

A relation $R$ is a congruential relation over $\lambda Y$-terms if:

\[
(M, N) \in R \quad \frac{M_1 \ R \ M_2}{M \ R \ N} \quad \frac{M_1 \ R \ M_2}{M_1 N \ R \ M_2 N} \quad \frac{M_1 \ R \ M_2}{NM_1 \ R \ NM_2} \quad \frac{M_1 \ R \ M_2}{\lambda x. M_1 \ R \ \lambda x. M_2}
\]

- $\beta$-contraction $\rightarrow_\beta$ is the least congruential relation s.t.:
  \[(\lambda x. M)N \rightarrow_\beta M[N/x]\]

- $\eta$-contraction $\rightarrow_\eta$ is the least congruential relation s.t.:
  \[\lambda x. Mx \rightarrow_\eta M \text{ when } x \notin FV(M),\]

- $\delta$-contraction $\rightarrow_\delta$ is the least congruential relation s.t.:
  \[YM \rightarrow_\delta M(YM)\]

- $\beta$-reduction $\rightarrow^*_\beta$: reflexive and transitive closure of $\rightarrow_\beta$,

- $\beta$-conversion $=^\beta$: reflexivity, transitive and symmetric closure of $\rightarrow_\beta$,

- $\beta\delta$-contraction $\rightarrow_{\beta\delta}$: union of $\rightarrow_\beta$ and $\rightarrow_\delta$,

- $\beta\delta$-reduction $\rightarrow^*_{\beta\delta}$: reflexive and transitive closure of $\rightarrow_{\beta\delta}$,
Operational semantics

A relation $\mathcal{R}$ is a congruential relation over $\lambda Y$-terms if:

$$
(M, N) \in \mathcal{R} \quad \frac{M_1 \mathcal{R} M_2}{M_1 N \mathcal{R} M_2 N} \quad \frac{M_1 \mathcal{R} M_2}{N M_1 \mathcal{R} N M_2} \quad \frac{M_1 \mathcal{R} M_2}{\lambda x. M_1 \mathcal{R} \lambda x. M_2}
$$

- $\beta$-contraction $\rightarrow_\beta$ is the least congruential relation s.t.:
  $$(\lambda x. M)N \rightarrow_\beta M[N/x]$$
- $\eta$-contraction $\rightarrow_\eta$ is the least congruential relation s.t.:
  $\lambda x. Mx \rightarrow_\eta M$ when $x \notin FV(M)$,
- $\delta$-contraction $\rightarrow_\delta$ is the least congruential relation s.t.:
  $YM \rightarrow_\delta M(YM)$.

- $\beta$-reduction $\rightarrow^*_\beta$: reflexive and transitive closure of $\rightarrow_\beta$,
- $\beta$-conversion $\equiv_\beta$: reflexivity, transitive and symmetric closure of $\rightarrow_\beta$,
- $\beta\delta$-contraction $\rightarrow_{\beta\delta}$: union of $\rightarrow_\beta$ and $\rightarrow_\delta$,
- $\beta\delta$-reduction $\rightarrow^*_\beta\delta$: reflexive and transitive closure of $\rightarrow_{\beta\delta}$,
- $\beta\delta$-conversion $\equiv_{\beta\delta}$: reflexive, transitive and symmetric closure of $\rightarrow_{\beta\delta}$. 
\( \eta \)-long forms

\( \eta \)-reduction cannot be considered as a computational rule. Indeed the proviso \( x \not\in FV(M) \) makes the rules non-local (this causes some trouble when working with infinitary calculi, c.f. \( \lambda x^0. Y(\lambda f^0 \to^0 y^0 . a(fx)y)x \)).
\(\eta\)-long forms

\(\eta\)-reduction cannot be considered as a computational rule. Indeed the proviso \(x \notin FV(M)\) makes the rules non-local (this causes some trouble when working with infinitary calculi, c.f. 
\[\lambda x^0. Y(\lambda f^0 \rightarrow^0 y^0 . a(fx)y)x.\]

\(\eta\)-long forms are ways of working modulo \(\eta\)-reduction without ever performing \(\eta\)-contractions. They also give some nice syntactic properties to \(\lambda\)-terms.

A term \(M\) is in \(\eta\)-long form when for every context \(C[]\) and term \(N\) so that \(C[N] = M\), if \(N\) has type \(A \rightarrow B\), then either \(N = \lambda x^A.P\) or \(C[] = C'[][]P\).

<table>
<thead>
<tr>
<th>Examples</th>
<th>Co-examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\lambda x^0.x^0)</td>
<td>(Y^0)</td>
</tr>
<tr>
<td>(\lambda f^0 \rightarrow^0 y^0 . a(fx)y)</td>
<td>(\lambda f^0 \rightarrow^0 . Y^0 f^0 \rightarrow^0)</td>
</tr>
</tbody>
</table>
Basic properties of simply typed $\lambda Y$-calculus

Theorem
For every term $M$, there is a term $M'$ in $\eta$-long form s.t. $M' \xrightarrow{\eta}^*$.

Theorem
If $M$ is in $\eta$-long form and $M \xrightarrow{\beta \delta}^* N$ then $N$ is in $\eta$-long form.

Theorem (Typing invariance (subject reduction))
If $M$ has type $A$ and $M \xrightarrow{\beta \delta \eta}^* N$ then $N$ has type $A$.

Theorem (Church-Rosser)
If $M \xrightarrow{\beta \delta}^* N_1$ and $M \xrightarrow{\beta \delta}^* N_2$ then there is $N$ so that $N_1 \xrightarrow{\beta \delta}^* N$ and $N_2 \xrightarrow{\beta \delta}^* N$.

Theorem (Strong normalization)
There is no infinite chain $M_0, \ldots M_n, \ldots$ so that for every $k$, $M_k \xrightarrow{\beta} M_{k+1}$. 
Typing invariance is true only because we work with terms typed à la Church.

\[ \rightarrow^{*}_{\beta\delta} \text{ does not satisfy the strong normalization property:} \]

- \[ Y^0(\lambda x^0.x^0) \rightarrow_{\delta} (\lambda x.x)(Y(\lambda x.x)) \rightarrow_{\beta} Y(\lambda x.x) \cdots, \]

- \[ Y^0(\lambda x^0.\mathit{a} x^0 x^0) \rightarrow_{\delta} (\lambda x.a x x)Y(\lambda x.a x x) \rightarrow_{\beta} \mathit{a}(Y(\lambda x.a x x))(Y(\lambda x.a x x)) \cdots \]

Working habit we usually consider terms in \( \eta \)-long form.
Böhm tree for $\Lambda Y$

Böhm trees are a sort of infinite normal form for $\Lambda Y$-terms.
Böhm tree for $\Lambda Y$

Böhm trees are a sort of infinite normal form for $\Lambda Y$-terms

If $M$ $\beta\delta$-reduces to $\lambda x_1 \ldots x_n. h M_1 \ldots M_n$:

$$BT(M) = \lambda x_1 \ldots x_n. h$$

$$BT(M_1) \ldots BT(M_n)$$
Böhm tree for $\Lambda Y$

Böhm trees are a sort of infinite normal form for $\Lambda Y$-terms

If $M \beta\delta$-reduces to $\lambda x_1 \ldots x_n.h M_1 \ldots M_n$:

$$BT(M) = \lambda x_1 \ldots x_n.h$$

otherwise:

$$BT(M) = \omega$$
Böhm trees are a sort of infinite normal form for $\Lambda Y$-terms

If $M$ $\beta\delta$-reduces to $\lambda x_1 \ldots x_n . h M_1 \ldots M_n$:

$$BT(M) = \lambda x_1 \ldots x_n . h$$

$$BT(M_1) \quad \cdots \quad BT(M_n)$$

otherwise:

$$BT(M) = \omega$$

When $M$ is closed and of type 0, $BT(M)$ is an infinite tree: a higher-order tree.
Ong’s Theorem

Theorem

Higher-order trees have decidable MSOL theories.
Ong’s Theorem

Theorem

Higher-order trees have decidable MSOL theories.

Plan of the proof:

- reduce the acceptability of a higher-order tree by a parity automaton to a (infinite) parity game on the execution of an abstract evaluation machine of the $\lambda Y$-calculus,
- construct an equivalent finite parity game.
Krivine Abstract Machine (KAM)

KAM is an abstract machine that we use so as to construct the Böhm tree of a \( \lambda Y \)-term. It is based on the notions of environment and of closures:

- \( \text{env} ::= \emptyset \mid \text{env}[x^A := Cl^A] \),
- \( Cl^A ::= (N, env) \) where \( N \in \Lambda_A \).

**Environments**: they map variables to their *values* that are represented as closures. Notice that closures are typed and that environments map variables of type \( A \) to closures of type \( A \).

**Closures**: they are formed by a pair whose first component is a term and whose second component is an environment that maps its free variables to their values.
KAM computation rules

- env ::= ∅ | env[xA := C1A],
- C1A ::= (N, env) where N ∈ ΛA.

A configuration of a KAM is a triple (M, ρ, S) where: M is a λY-term, ρ is an environment, S is a stack (i.e. a sequence of closures).

- (MN, ρ, S) → (M, ρ, (N, ρ)S)
- (λx.M, ρ, Cl S) → (M, ρ[x := Cl], S)
- (YN, ρ, S) → (N(YN), ρ, S)
- (x, ρ, S) → (N, ρ′, S) if ρ(x) = (N, ρ′)

Notice that the evaluation is deterministic.
KAM computation rules

- env ::= ∅ | env[x^A := C^A],
- C^A ::= (N, env) where N ∈ Λ_A.

A configuration of a KAM is a triple (M, ρ, S) where: M is a λY-term, ρ is an environment, S is a stack (i.e. a sequence of closures).

\[
\begin{align*}
(MN, ρ, S) & \rightarrow (M, ρ, (N, ρ)S) \\
(λx.M, ρ, Cl S) & \rightarrow (M, ρ[x := Cl], S) \\
(YN, ρ, S) & \rightarrow (N(YN), ρ, S) \\
(x, ρ, S) & \rightarrow (N, ρ', S) \text{ if } ρ(x) = (N, ρ')
\end{align*}
\]

Notice that the evaluation is deterministic.
A closure \((N, \rho)\) is *closed* when for every \(x\) that has a free occurrence in \(N\), \(\rho(x)\) is defined and is a *closed* closure.

A configuration \((M, \rho, Cl_1 \ldots Cl_n)\) is *well-typed and has type* \(A\) when \((M, \rho)\) is closed, \(M\) has type \(A_1 \rightarrow \cdots \rightarrow A_n \rightarrow A\) and the closures \(Cl_1, \ldots, Cl_n\) are closed and respectively of type \(A_1, \ldots, A_n\).

**Lemma**

*If \((M, \rho, S)\) is a well-typed configuration of type \(A\) and \((M, \rho, S) \rightarrow (M', \rho', S')\), then \((M', \rho', S')\) is a well-typed configuration of type \(A\).***
KAM and reduction

Given a closure $Cl = (N, \rho)$, we define $\overline{Cl}$ to be the $\lambda Y$-term: $N[\rho(x_1)/x_1, \ldots, \rho(x_n)/x_n]$ where $FV(N) = \{x_1, \ldots, x_n\}$.

For a configuration $C = (M, \rho, Cl_1 \ldots Cl_n)$, we associate the term $\overline{C} = (M, \rho)\overline{Cl_1} \ldots \overline{Cl_n}$.

**Lemma**

*If $C_1$ and $C_2$ are two configurations of the KAM and $C_1 \rightarrow C_2$, then $\overline{C_1} \rightarrow_{\beta\delta} \overline{C_2}$.***
KAM and reduction

Given a closure $\mathcal{C}l = (N, \rho)$, we define $\overline{\mathcal{C}l}$ to be the $\lambda Y$-term: $N[\rho(x_1)/x_1, \ldots, \rho(x_n)/x_n]$ where $FV(N) = \{x_1, \ldots, x_n\}$.

For a configuration $C = (M, \rho, \mathcal{C}l_1 \ldots \mathcal{C}l_n)$, we associate the term $\overline{C} = (M, \rho)\overline{\mathcal{C}l_1} \ldots \overline{\mathcal{C}l_n}$.

**Lemma**

*If $C_1$ and $C_2$ are two configurations of the KAM and $C_1 \rightarrow C_2$, then $\overline{C_1} \xrightarrow{\star} \overline{C_2}$."

Actually, we could be more precise: the KAM is actually performing head reduction. It reduces the leftmost redices in terms of the form $(YM)M_1 \ldots M_n$, $(\lambda x.M)NM_1 \ldots M_n$. 
KAM and Böhm tree

With the KAM, we may define a tree, the Krivine tree of a configuration $C$, $Kt(C)$:

- if $C$ reduces to $(\lambda x. N, \rho, \varepsilon)$ then $Kt(C)$ has a root labeled $\lambda x.$ and its child is $Kt(N, \rho, \varepsilon)$,
- if $C$ reduces to $(x, \rho, (N_1, \rho_1) \ldots (N_n, \rho_n))$ and $x$ is not in the domain of $\rho$, then the root of $Kt(C)$ is $x$ and its daughters are $Kt(N_1, \rho_1), \ldots Kt(N_n, \rho_n),$
- if $C$ reduces to $(a, \rho, (N_1, \rho_1)(N_2, \rho_2))$, then the root of $Kt(C)$ is $a$ and its daughters are $Kt(N_1, \rho_1), Kt(N_2, \rho_2),$
- in the other cases, $Kt(C) = \omega^A$ when $C$ is of type $A$.

For a given $\lambda Y$-term $Kt(M, \emptyset, \varepsilon)$ is also called the Lévy-Longo tree of $M$. 
Theorem

If $M$ is closed term of type 0, $Kt(M, \emptyset, \varepsilon)$ is isomorphic to $BT(M)$. 
Theorem

If $M$ is closed term of type 0, $Kt(M, \emptyset, \varepsilon)$ is isomorphic to $BT(M)$. The proof is a simpler version of the standardization Theorem. It suffices to show that a term $M$ can be reduced to a term of the form $\lambda x_1 \ldots x_m. hN_1 \ldots N_n$ iff head-reduction reduces it to a term of the form $\lambda x_1 \ldots x_m. hP_1 \ldots P_n$. 
Böhm trees and parity automata

Definition
A parity automaton is a tuple $A = (Q, \Sigma, \delta : Q \times \Sigma \rightarrow \mathcal{P}(Q^2), rk : Q \rightarrow \mathbb{N})$. A run of $A$ on a Böhm tree is accepting from a state $q^0$ iff:

- the root is labeled with state $q^0$,
- in every infinite branch the parity condition is satisfied,
- in every finite branch (those that end with $\omega^0$ as a leaf) the state $q$ at the leaf verifies that $rk(q)$ is even.
The game $\mathcal{K}(A, M, q^0)$ (1)

We fix a parity automaton $A = (Q, \Sigma, \delta : Q \times \Sigma \to \mathcal{P}(Q^2), rk : Q \to \mathbb{N})$ and a closed term $M$ of type 0. We now define a parity game $\mathcal{K}(A, M, q^0)$: whose positions are a set $V$ labeled with $M$-enriched configurations of the KAM:

$M$-enriched closures are of the form $(v, N, \rho)$ where:

- $v$ is an element of $V$ (the node where the closure has been created),
The game $\mathcal{K}(\mathcal{A}, M, q^0)$ (1)

We fix a parity automaton $\mathcal{A} = (Q, \Sigma, \delta : Q \times \Sigma \to \mathcal{P}(Q^2), \text{rk} : Q \to \mathbb{N})$ and a closed term $M$ of type 0. We now define a parity game $\mathcal{K}(\mathcal{A}, M, q^0)$: whose positions are a set $V$ labeled with $M$-enriched configurations of the KAM:

$M$-enriched closures are of the form $(v, N, \rho)$ where:

- $v$ is an element of $V$ (the node where the closure has been created),
- either $N$ is a subterm of $M$, or $N = P(YP)$ where $YP$ is a subterm of $M$,
The game $K(\mathcal{A}, M, q^0)$ (1)

We fix a parity automaton $\mathcal{A} = (Q, \Sigma, \delta : Q \times \Sigma \to \mathcal{P}(Q^2), rk : Q \to \mathbb{N})$ and a closed term $M$ of type 0. We now define a parity game $K(\mathcal{A}, M, q^0)$: whose positions are a set $V$ labeled with $M$-enriched configurations of the KAM:

$M$-enriched closures are of the form $(v, N, \rho)$ where:

- $v$ is an element of $V$ (the node where the closure has been created),
- either $N$ is a subterm of $M$, or $N = P(YP)$ where $YP$ is a subterm of $M$,
- an $M$-enriched environment, i.e. a mapping from variable to $M$-enriched closures of the corresponding type.
The game $\mathcal{K}(A, M, q^0)$ (1)

We fix a parity automaton $A = (Q, \Sigma, \delta : Q \times \Sigma \rightarrow \mathcal{P}(Q^2), rk : Q \rightarrow \mathbb{N})$ and a closed term $M$ of type 0. We now define a parity game $\mathcal{K}(A, M, q^0)$: whose positions are a set $V$ labeled with $M$-enriched configurations of the KAM:

$M$-enriched closures are of the form $(v, N, \rho)$ where:

- $v$ is an element of $V$ (the node where the closure has been created),
- either $N$ is a subterm of $M$, or $N = P(YP)$ where $YP$ is a subterm of $M$,
- an $M$-enriched environment, i.e. a mapping from variable to $M$-enriched closures of the corresponding type.

An $M$-enriched configuration is a triple $(N, \rho, S) : q$ where:

- either $N$ is a subterm of $M$, or $N = P(YP)$ where $YP$ is a subterm of $M$,
The game $\mathcal{K}(\mathcal{A}, M, q^0)$ (1)

We fix a parity automaton $\mathcal{A} = (Q, \Sigma, \delta : Q \times \Sigma \rightarrow \mathcal{P}(Q^2), rk : Q \rightarrow \mathbb{N})$ and a closed term $M$ of type 0. We now define a parity game $\mathcal{K}(\mathcal{A}, M, q^0)$: whose positions are a set $V$ labeled with $M$-enriched configurations of the KAM:

$M$-enriched configurations are of the form $(v, N, \rho)$ where:

- $v$ is an element of $V$ (the node where the closure has been created),
- either $N$ is a subterm of $M$, or $N = P(YP)$ where $YP$ is a subterm of $M$,
- an $M$-enriched environment, i.e. a mapping from variable to $M$-enriched closures of the corresponding type.

An $M$-enriched configuration is a triple $(N, \rho, S) : q$ where:

- either $N$ is a subterm of $M$, or $N = P(YP)$ where $YP$ is a subterm of $M$,
- $\rho$ is an $M$-enriched environment,
The game $\mathcal{K}(A, M, q^0)$ (1)

We fix a parity automaton $A = (Q, \Sigma, \delta : Q \times \Sigma \to \mathcal{P}(Q^2), \text{rk} : Q \to \mathbb{N})$ and a closed term $M$ of type 0. We now define a parity game $\mathcal{K}(A, M, q^0)$: whose positions are a set $V$ labeled with $M$-enriched configurations of the KAM:

$M$-enriched closures are of the form $(v, N, \rho)$ where:

- $v$ is an element of $V$ (the node where the closure has been created),
- either $N$ is a subterm of $M$, or $N = P(YP)$ where $YP$ is a subterm of $M$,
- an $M$-enriched environment, i.e. a mapping from variable to $M$-enriched closures of the corresponding type.

An $M$-enriched configuration is a triple $(N, \rho, S) : q$ where:

- either $N$ is a subterm of $M$, or $N = P(YP)$ where $YP$ is a subterm of $M$,
- $\rho$ is an $M$-enriched environment,
- $S$ is a sequence of $M$-enriched closures,
The game $\mathcal{K}(\mathcal{A}, M, q^0)$ (1)

We fix a parity automaton $\mathcal{A} = (Q, \Sigma, \delta : Q \times \Sigma \to \mathcal{P}(Q^2), \text{rk} : Q \to \mathbb{N})$ and a closed term $M$ of type 0. We now define a parity game $\mathcal{K}(\mathcal{A}, M, q^0)$: whose positions are a set $V$ labeled with $M$-enriched configurations of the KAM:

$M$-enriched closures are of the form $(v, N, \rho)$ where:

- $v$ is an element of $V$ (the node where the closure has been created),
- either $N$ is a subterm of $M$, or $N = P(YP)$ where $YP$ is a subterm of $M$,
- an $M$-enriched environment, i.e. a mapping from variable to $M$-enriched closures of the corresponding type.

An $M$-enriched configuration is a triple $(N, \rho, S) : q$ where:

- either $N$ is a subterm of $M$, or $N = P(YP)$ where $YP$ is a subterm of $M$,
- $\rho$ is an $M$-enriched environment,
- $S$ is a sequence of $M$-enriched closures,
- $q$ is a state of $\mathcal{A}$. 
The game $\mathcal{K}(A, M, q^0)$ (2)

Positions in $\mathcal{K}(A, M)$ are labeled:

- either with $M$-enriched configurations,
- or elements of the form: $(q_1, q_2), (a, \rho, Cl_1 Cl_2) : q$ where $\rho$ is an $M$-enriched environment and $Cl_1$ and $Cl_2$ are $M$-enriched closures.
The game $\mathcal{K}(A, M, q^0)$ (3)

The positions in $\mathcal{K}(A, M, q^0)$ are related to each other by reductions of the KAM:

- the root of the game is labeled $(M, \emptyset, \varepsilon): q^0$, 

...
The game \( \mathcal{K}(A, M, q^0) \) (3)

The positions in \( \mathcal{K}(A, M, q^0) \) are related to each other by reductions of the KAM:

- the root of the game is labeled \((M, \emptyset, \varepsilon) : q^0,\)
- if the node \( v \) is labeled with \((N_1 N_2, \rho, S) : q,\) then it has a unique successor labeled \((N_1, \rho, (v, N_2, \rho)S) : q,\)
The game $\mathcal{K}(A, M, q^0)$ (3)

The positions in $\mathcal{K}(A, M, q^0)$ are related to each other by reductions of the KAM:

- the root of the game is labeled $(M, \emptyset, \varepsilon) : q^0$,
- if the node $v$ is labeled with $(N_1 N_2, \rho, S) : q$, then it has a unique successor labeled $(N_1, \rho, (v, N_2, \rho) S) : q$,
- if a node is labeled $(\lambda x. N, \rho, Cl S) : q$, then it has a unique successor labeled $(N, \rho[x := Cl], S) : q$,
The game $\mathcal{K}(A, M, q^0)$ (3)

The positions in $\mathcal{K}(A, M, q^0)$ are related to each other by reductions of the KAM:

- the root of the game is labeled $(M, \emptyset, \varepsilon) : q^0$,
- if the node $v$ is labeled with $(N_1 N_2, \rho, S) : q$, then it has a unique successor labeled $(N_1, \rho, (v, N_2, \rho)S) : q$,
- if a node is labeled $(\lambda x. N, \rho, Cl S) : q$, then it has a unique successor labeled $(N, \rho[x := Cl], S) : q$,
- if a node is labeled $(YN, \rho, S) : q$, then it has a unique successor labeled $(N(YN), \rho, S) : q$,
The game $\mathcal{K}(\mathcal{A}, M, q^0)$ (3)

The positions in $\mathcal{K}(\mathcal{A}, M, q^0)$ are related to each other by reductions of the KAM:

- the root of the game is labeled $(M, \emptyset, \varepsilon) : q^0$,
- if the node $v$ is labeled with $(N_1 N_2, \rho, S) : q$, then it has a unique successor labeled $(N_1, \rho, (v, N_2, \rho)S) : q$,
- if a node is labeled $(\lambda x. N, \rho, Cl S) : q$, then it has a unique successor labeled $(N, \rho[x := Cl], S) : q$,
- if a node is labeled $(YN, \rho, S) : q$, then it has a unique successor labeled $(N(YN), \rho, S) : q$,
- if a node is labeled $(x, \rho, S) : q$ and $\rho(x) = (v, N, \rho')$, then it has a unique successor labeled $(N, \rho', S) : q$. 


The game $\mathcal{K}(A, M, q^0)$ (3)

The positions in $\mathcal{K}(A, M, q^0)$ are related to each other by reductions of the KAM:

- the root of the game is labeled $(M, \emptyset, \varepsilon) : q^0$,
- if the node $v$ is labeled with $(N_1N_2, \rho, S) : q$, then it has a unique successor labeled $(N_1, \rho, (v, N_2, \rho)S) : q$,
- if a node is labeled $(\lambda x.N, \rho, Cl S) : q$, then it has a unique successor labeled $(N, \rho[x := Cl], S) : q$,
- if a node is labeled $(YN, \rho, S) : q$, then it has a unique successor labeled $(N(YN), \rho, S) : q$,
- if a node is labeled $(x, \rho, S) : q$ and $\rho(x) = (v, N, \rho')$, then it has a unique successor labeled $(N, \rho', S) : q$,
- if a node is labeled $(a, \rho, Cl_1 Cl_2) : q$ then for every $(q_1, q_2)$ in $\delta(a, q)$ it has a successor $(q_1, q_2), (a, \rho, Cl_1 Cl_2) : q$, 
The game $\mathcal{K}(A, M, q^0)$ (3)

The positions in $\mathcal{K}(A, M, q^0)$ are related to each other by reductions of the KAM:

- the root of the game is labeled $(M, \emptyset, \varepsilon) : q^0$,
- if the node $v$ is labeled with $(N_1 N_2, \rho, S) : q$, then it has a unique successor labeled $(N_1, \rho, (v, N_2, \rho)S) : q$,
- if a node is labeled $(\lambda x. N, \rho, Cl S) : q$, then it has a unique successor labeled $(N, \rho[x := Cl], S) : q$,
- if a node is labeled $(YN, \rho, S) : q$, then it has a unique successor labeled $(N(YN), \rho, S) : q$,
- if a node is labeled $(x, \rho, S) : q$ and $\rho(x) = (v, N, \rho')$, then it has a unique successor labeled $(N, \rho', S) : q$,
- if a node is labeled $(a, \rho, Cl_1 Cl_2) : q$ then for every $(q_1, q_2)$ in $\delta(a, q)$ it has a successor $(q_1, q_2), (a, \rho, Cl_1 Cl_2) : q$,
- if a node is labeled $(q_1, q_2), (a, \rho, (N_1, \rho_1)(N_2, \rho_2)) : q$, then it has two successors $(N_1, \rho_1, \varepsilon) : q_1$ and $(N_2, \rho_2, \varepsilon) : q_2$. 
The game $\mathcal{K}(A, M, q^0)$ (4)

The only positions where there is some choice to be made are the positions labeled:

- $(a, \rho, Cl_1 Cl_2) : q$ which belong to Eve,
- $(q_1, q_2), (a, \rho, Cl_1 Cl_2) : q$ which belong to Adam.
The game $\mathcal{K}(A, M, q^0)$ (4)

The only positions where there is some choice to be made are the positions labeled:

- $(a, \rho, Cl_1 Cl_2) : q$ which belong to Eve,
- $(q_1, q_2), (a, \rho, Cl_1 Cl_2) : q$ which belong to Adam.
- Each position has as rank that of its state.
The game $\mathcal{K}(\mathcal{A}, M, q^0)$ (4)

The only positions where there is some choice to be made are the positions labeled:

- $(a, \rho, Cl_1 Cl_2) : q$ which belong to Eve,
- $(q_1, q_2), (a, \rho, Cl_1 Cl_2) : q$ which belong to Adam.

- Each position has as rank that of its state.
- Eve looses when she cannot use a transition of $\mathcal{A}$: she looses on finite path of the game.
The game $\mathcal{K}(\mathcal{A}, M, q^0)$ (4)

The only positions where there is some choice to be made are the positions labeled:

- $(a, \rho, Cl_1 Cl_2) : q$ which belong to Eve,
- $(q_1, q_2), (a, \rho, Cl_1 Cl_2) : q$ which belong to Adam.

- Each position has as rank that of its state.
- Eve looses when she cannot use a transition of $\mathcal{A}$: she looses on finite path of the game.
- Eve wins a play iff on the infinite play the parity condition is satisfied.

**Theorem**

_Eve wins in $\mathcal{K}(\mathcal{A}, M, q^0)$ iff $\mathcal{A}$ accepts $BT(M)$ from state $q^0$. _
Towards a finite parity game

We are going to construct a finite parity game $G(\mathcal{A}, M, q^0)$ that is equivalent to $K(\mathcal{A}, M, q^0)$. For this we are going to express some invariants using standard model of $\lambda$-calculus, residuals:

$R_0 = \mathcal{P}(\{ (q, i) \mid rk(q) \leq i \leq m \})$ with
$m = \max\{ rk(q) \mid q \in Q \}$.
Towards a finite parity game

We are going to construct a finite parity game $G(\mathcal{A}, M, q^0)$ that is equivalent to $K(\mathcal{A}, M, q^0)$. For this we are going to express some invariants using standard model of λ-calculus, residuals:

- $R_0 = \mathcal{P}(\{(q, i) \mid rk(q) \leq i \leq m\})$ with $m = \max\{rk(q) \mid q \in Q\}$.
- $R_{A \to B} = R_B^{\mathcal{R}_A}$. 

Given a residual $R$, and $0 \leq k \leq m$, we define the residual $R \Downarrow k$ to be the residual so that for every $S_1, \ldots, S_n$:

\[ R \Downarrow k(S_1, \ldots, S_n) = \{(q, i) \mid (q, k) \in R(S_1, \ldots, S_n) \land 0 \leq i \leq k\} \cup \{(q, i) \mid (q, i) \in R(S_1, \ldots, S_n) \land k < i\} \]

A valuation is a partial mapping from variables to residuals of the right type. For a valuation $\nu$, we write $\nu \Downarrow k$ for the valuation s.t. $\nu \Downarrow k(x) = \nu(x) \Downarrow k$. 
Towards a finite parity game

We are going to construct a finite parity game $G(\mathcal{A}, M, q^0)$ that is equivalent to $\mathcal{K}(\mathcal{A}, M, q^0)$. For this we are going to express some invariants using standard model of $\lambda$-calculus, residuals:

- $\mathcal{R}_0 = \mathcal{P}(\{(q, i) \mid rk(q) \leq i \leq m\})$ with $m = \max\{rk(q) \mid q \in Q\}$.
- $\mathcal{R}_{A \rightarrow B} = \mathcal{R}_{B}^{\mathcal{R}_{A}}$.

Given a residual $R$, and $0 \leq k \leq m$, we define the residual $R \downharpoonright_k$ to be the residual so that for every $S_1, \ldots, S_n$:

$$R \downharpoonright_k(S_1, \ldots, S_n) = \{(q, i) \mid (q, k) \in R(S_1, \ldots, S_n) \land 0 \leq i \leq k\} \cup \{(q, i) \mid (q, i) \in R(S_1, \ldots, S_n) \land k < i\}$$
Towards a finite parity game

We are going to construct a finite parity game $G(A, M, q^0)$ that is equivalent to $K(A, M, q^0)$. For this we are going to express some invariants using standard model of $\lambda$-calculus, residuals:

- $R_0 = \mathcal{P}(\{(q, i) \mid rk(q) \leq i \leq m\})$ with $m = \max\{rk(q) \mid q \in Q\}$.
- $R_{A \rightarrow B} = R_B^R_A$.

Given a residual $R$, and $0 \leq k \leq m$, we define the residual $R\downharpoonright_k$ to be the residual so that for every $S_1, \ldots, S_n$:

$$R\downharpoonright_k(S_1, \ldots, S_n) = \{(q, i) \mid (q, k) \in R(S_1, \ldots, S_n) \land 0 \leq i \leq k\} \cup \{(q, i) \mid (q, i) \in R(S_1, \ldots, S_n) \land k < i\}$$

A valuation is a partial mapping from variables to residuals of the right type. For a valuation $\nu$, we write $\nu\downharpoonright_k$ for the valuation s.t. $\nu\downharpoonright_k(x) = \nu(x)\downharpoonright_k$. 
Lemma

For every residual, $R$ and $k_1$, $k_2$, $(R \downarrow_{k_1}) \downarrow_{k_2} = R \downarrow_{\max(k_1,k_2)}$.

Proof.
Exercice.

□
Lemma

For every residual, $R$ and $k_1, k_2$, $(R \downarrow_{k_1}) \downarrow_{k_2} = R \downarrow_{\max(k_1, k_2)}$.

Proof.

Exercice.

For a state $q$ we may write $R \downarrow_q$ for $R \downarrow_{rk(q)}$.  

\[]
The game $G(\mathcal{A}, M, q^0)$

The positions in the game $G(\mathcal{A}, M, q^0)$ are labeled by the forms

- $(N, \nu, \vec{R}) : q$, or
- $(q_1, q_2), (a, \nu, R_1 R_2) : q$ or
- $T, (N_1 N_2, \rho, \vec{R}) : q$

where

- $q, q_1, q_2$ are states of $\mathcal{A}$,
- $N, N_1, N_2$ is a subterm of $M$,
- $\nu$ is a valuation that maps variables to residuals of the corresponding type,
- $\vec{R}$ is a sequence of residuals $R_1 \ldots R_n$ respectively of type $A_1, \ldots, A_n$ when $N$ has type $A_1 \rightarrow \cdots \rightarrow A_n \rightarrow 0$,
- $R_1$ and $R_2$ are residuals from $\mathcal{R}_0$,
- $T$ is a residual with the same type as $N_2$. 
The rules of $G(\mathcal{A}, M, q^0)$

The positions in $G(\mathcal{A}, M, q^0)$ are related to each other by reductions of the KAM:

- the root of the game is labeled $(M, \emptyset, \varepsilon) : q^0$, 

The rules of $G(A, M, q^0)$

The positions in $G(A, M, q^0)$ are related to each other by reductions of the KAM:

- the root of the game is labeled $(M, \emptyset, \varepsilon) : q^0$,
- if a node is labeled $(\lambda x. N, \nu, T \vec{R}) : q$, then it has a unique successor labeled $(N, \nu[x := T], \vec{R}) : q$,.
The rules of \( G(\mathcal{A}, M, q^0) \)

The positions in \( G(\mathcal{A}, M, q^0) \) are related to each other by reductions of the KAM:

- ▶ the root of the game is labeled \((M, \emptyset, \varepsilon) : q^0\),
- ▶ if a node is labeled \((\lambda x. N, \nu, T \mathcal{R}) : q\), then it has a unique successor labeled \((N, \nu[x := T], \mathcal{R}) : q\),
- ▶ if a node is labeled \((YN, \nu, \mathcal{R}) : q\), then it has a unique successor labeled \((N(YN), \nu, S) : q\),
The rules of $G(A, M, q^0)$

The positions in $G(A, M, q^0)$ are related to each other by reductions of the KAM:

- the root of the game is labeled $(M, \emptyset, \varepsilon) : q^0$,
- if a node is labeled $(\lambda x. N, \nu, T \vec{R}) : q$, then it has a unique successor labeled $(N, \nu[x := T], \vec{R}) : q$,
- if a node is labeled $(YN, \nu, \vec{R}) : q$, then it has a unique successor labeled $(N(YN), \nu, S) : q$,
- if a node is labeled $(a, \rho, R_1 R_2) : q$ then for every $(q_1, q_2)$ in $\delta(a, q)$ it has a successor $(q_1, q_2), (a, \rho, R_1 R_2) : q$,
The rules of $\mathcal{G}(A, M, q^0)$

The positions in $\mathcal{G}(A, M, q^0)$ are related to each other by reductions of the KAM:

- the root of the game is labeled $(M, \emptyset, \varepsilon): q^0$,
- if a node is labeled $(\lambda x.N, \nu, T \vec{R}) : q$, then it has a unique successor labeled $(N, \nu[x := T], \vec{R}) : q$,
- if a node is labeled $(YN, \nu, \vec{R}) : q$, then it has a unique successor labeled $(N(YN), \nu, S) : q$,
- if a node is labeled $(a, \rho, R_1R_2) : q$ then for every $(q_1, q_2)$ in $\delta(a, q)$ it has a successor $(q_1, q_2), (a, \rho, R_1R_2) : q$,
- if a node is labeled $(N_1N_2, \nu, \vec{R}) : q$, then for every residual $T$ with the same type as $N_2$, $T, (N_1N_2, \nu, \vec{R}) : q$ is a successor of that node,
The rules of $G(\mathcal{A}, M, q^0)$

The positions in $G(\mathcal{A}, M, q^0)$ are related to each other by reductions of the KAM:

- the root of the game is labeled $(M, \emptyset, \varepsilon) : q^0$,
- if a node is labeled $(\lambda x.N, \nu, T \vec{R}) : q$, then it has a unique successor labeled $(N, \nu[x := T], \vec{R}) : q$,
- if a node is labeled $(YN, \nu, \vec{R}) : q$, then it has a unique successor labeled $(N(YN), \nu, S) : q$,
- if a node is labeled $(a, \rho, R_1 R_2) : q$ then for every $(q_1, q_2)$ in $\delta(a, q)$ it has a successor $(q_1, q_2), (a, \rho, R_1 R_2) : q$,
- if a node is labeled $(N_1 N_2, \nu, \vec{R}) : q$, then for every residual $T$ with the same type as $N_2$, $T, (N_1 N_2, \nu, \vec{R}) : q$ is a successor of that node,
- if a node is labeled $T, (N_1 N_2, \nu, \vec{R}) : q$, then its successors are:
  - $(N_1, \nu, T \downarrow_{rk(q)} \vec{R}) : q$,
  - $(N_2, \nu \downarrow_l, K_1 \ldots K_n) : q'$, for every residuals of the right type $K_1, \ldots, K_n$, and every $(q', l)$ in $T(K_1, \ldots, K_n)$. 
Winning conditions for $G(\mathcal{A}, M, q^0)$

The positions with possibly several successors respectively belong to:

- $(a, \rho, R_1 R_2) : q$ belong to Eve,
- $(N_1 N_2, \nu, \bar{R}) : q$ belong to Eve,
- $T, (N_1 N_2, \nu, \bar{R})$ belong to Adam.
Winning conditions for $G(\mathcal{A}, M, q^0)$

The positions with possibly several successors respectively belong to:

- $(a, \rho, R_1 R_2) : q$ belong to Eve,
- $(N_1 N_2, \nu, \tilde{R}) : q$ belong to Eve,
- $T, (N_1 N_2, \nu, \tilde{R})$ belong to Adam.

The positions $(x, \nu, \tilde{R}) : q$ and $(q_1, q_2), (a, \rho, R_1 R_2) : q$ have no successors:

- in position $(x, \nu, \tilde{R}) : q$ Eve wins iff $(q, rk(q)) \in \nu(x)(\tilde{R}),$
- in position $(q_1, q_2), (a, \rho, R_1 R_2) : q$ Eve wins iff $(q_i, rk(q_i))$ is in $R_i \sqsubset_q \Delta_i.$
Winning conditions for $G(\mathcal{A}, M, q^0)$

The positions with possibly several successors respectively belong to:

- $(a, \rho, R_1 R_2) : q$ belong to Eve,
- $(N_1 N_2, \nu, \vec{R}) : q$ belong to Eve,
- $T, (N_1 N_2, \nu, \vec{R})$ belong to Adam.

The positions $(x, \nu, \vec{R}) : q$ and $(q_1, q_2), (a, \rho, R_1 R_2) : q$ have no successors:

- in position $(x, \nu, \vec{R}) : q$ Eve wins iff $(q, rk(q)) \in \nu(x)(\vec{R})$,
- in position $(q_1, q_2), (a, \rho, R_1 R_2) : q$ Eve wins iff $(q_i, rk(q_i))$ is in $R_i \mid_{q_i} q_i$.

In the game $G(\mathcal{A}, M, q^0)$ the ranks are on transitions rather than on positions:

- on deterministic transitions, the state remains unchanged and the rank of the transition is that of that state,
Winning conditions for $G(\mathcal{A}, M, q^0)$

The positions with possibly several successors respectively belong to:

- $(a, \rho, R_1 R_2) : q$ belong to Eve,
- $(N_1 N_2, \nu, \bar{R}) : q$ belong to Eve,
- $T, (N_1 N_2, \nu, \bar{R})$ belong to Adam.

The positions $(x, \nu, \bar{R}) : q$ and $(q_1, q_2), (a, \rho, R_1 R_2) : q$ have no successors:

- in position $(x, \nu, \bar{R}) : q$ Eve wins iff $(q, rk(q)) \in \nu(x)(\bar{R})$,
- in position $(q_1, q_2), (a, \rho, R_1 R_2) : q$ Eve wins iff $(q_i, rk(q_i))$ is in $R_i \downarrow_q \downarrow_{q_i}$.

In the game $G(\mathcal{A}, M, q^0)$ the ranks are on transitions rather than on positions:

- on deterministic transitions, the state remains unchanged and the rank of the transition is that of that state,
- the rank of a transition from the position $T, (N_1 N_2, \nu, \bar{R}) : q$ to the position $(N_1, \nu, T \downarrow_{rk(q)} \bar{R}) : q$ is $rk(q)$.\[\]
Winning conditions for $\mathcal{G}(\mathcal{A}, M, q^0)$

The positions with possibly several successors respectively belong to:
- $(a, \rho, R_1 R_2) : q$ belong to Eve,
- $(N_1 N_2, \nu, \vec{R}) : q$ belong to Eve,
- $T, (N_1 N_2, \nu, \vec{R})$ belong to Adam.

The positions $(x, \nu, \vec{R}) : q$ and $(q_1, q_2), (a, \rho, R_1 R_2) : q$ have no successors:
- in position $(x, \nu, \vec{R}) : q$ Eve wins iff $(q, rk(q)) \in \nu(x)(\vec{R}),$
- in position $(q_1, q_2), (a, \rho, R_1 R_2) : q$ Eve wins iff $(q_i, rk(q_i))$ is in $R_i \downarrow_q \downarrow_{q_i}.$

In the game $\mathcal{G}(\mathcal{A}, M, q^0)$ the ranks are on transitions rather than on positions:
- on deterministic transitions, the state remains unchanged and the rank of the transition is that of that state,
- the rank of a transition from the position $T, (N_1 N_2, \nu, \vec{R}) : q$ to the position $(N_1, \nu, T \downarrow_{rk(q)} \vec{R}) : q$ is $rk(q),$
- the transition from the position $T, (N_1 N_2, \nu, \vec{R}) : q$ to the position $(N_2, \nu \downarrow_l, K_1 \ldots K_n) : q',$ for $(q', l)$ in $T(K_1, \ldots, K_n)$ is $l.$

On infinite plays Eve wins iff the parity condition is satisfied.
Theorem

*Eve has a winning strategy on* $\mathcal{K}(A, M, q^0)$ *iff she has a winning strategy on* $\mathcal{G}(A, M, q^0)$. 

Proof.

The proof relies on the fact that parity games are determined:

- Either Eve has a winning strategy on $\mathcal{K}(A, M, q^0)$.
- Or Adam has a winning strategy on $\mathcal{K}(A, M, q^0)$.

The proof is in two steps:

- If Eve has a winning strategy on $\mathcal{K}(A, M, q^0)$ then she has a winning strategy on $\mathcal{G}(A, M, q^0)$.
- If Adam has a winning strategy on $\mathcal{K}(A, M, q^0)$ then he has a winning strategy on $\mathcal{G}(A, M, q^0)$. 

Equivalence of $\mathcal{K}(A, M, q^0)$ and $\mathcal{G}(A, M, q^0)$

**Theorem**

*Eve has a winning strategy on $\mathcal{K}(A, M, q^0)$ iff she has a winning strategy on $\mathcal{G}(A, M, q^0)$.*

**Proof.**

The proof relies on the fact that parity games are determined:

- either Eve has a winning strategy on $\mathcal{K}(A, M, q^0)$
- or Adam has a winning strategy on $\mathcal{K}(A, M, q^0)$.
Theorem

*Eve has a winning strategy on* $\mathcal{K}(A, M, q^0)$ *iff she has a winning strategy on* $\mathcal{G}(A, M, q^0)$.

Proof.

The proof relies on the fact that parity games are determined:

- either Eve has a winning strategy on $\mathcal{K}(A, M, q^0)$
- or Adam has a winning strategy on $\mathcal{K}(A, M, q^0)$.

The proof is in two steps:

- if Eve has a winning strategy on $\mathcal{K}(A, M, q^0)$ then she has a winning strategy on $\mathcal{G}(A, M, q^0)$,
- if Adam has a winning strategy on $\mathcal{K}(A, M, q^0)$ then he has a winning strategy on $\mathcal{G}(A, M, q^0)$. 

\[\Box\]